



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/670,898	09/25/2003	William J. Masck	LOT920030024US1	5987
23550	7590	11/17/2006	EXAMINER: [REDACTED]	
HOFFMAN WARNICK & D'ALESSANDRO, LLC 75 STATE STREET 14TH FLOOR ALBANY, NY 12207			MITCHELL, JASON D	
			ART UNIT	PAPER NUMBER
			2193	

DATE MAILED: 11/17/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	10/670,898	MASEK ET AL.
	Examiner	Art Unit
	Jason Mitchell	2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 20 September 2006.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-26 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-26 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner. Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a). Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All
 - b) Some *
 - c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) Notice of Informal Patent Application
- 6) Other: _____

DETAILED ACTION

This action is in response to remarks filed 9/20/06.

At Applicant's request, claims 6, 15 and 24 have been amended. Claims 1-26 are pending in this application.

Response to Arguments/Amendments

Remark on the Specification.

Applicant's Amendments to were sufficient to overcome the objection to the specification based on the use of trademarks. Accordingly that objection has been withdrawn.

Rejection of claims 9 and 18 under 35 USC 112 2nd

Applicant's statement indicating "applications that do not meet the reentrancy requirements are not subject matter that is claimed" are taken to imply that the claimed system and program product are not designed to be applied to 'non-reentrant' applications, and the corresponding rejections has been withdrawn.

Further, in light of Applicant's statement that "reentrancy requirements", as claimed, "would be readily understood by one of ordinary skill in the art" the rejection upon these grounds will be withdrawn. Further, as Applicant has failed to present timely arguments indicating Examiner has misinterpreted the phrase in the prior art rejection, it is assumed that the term is used to indicate any reentrant code and that the "reentrant nature" of Duggan's system (e.g. col. 21, lines 57-61) anticipates this limitation.

Rejection of Claims 1-5, 7-14, 16-23 and 25-26 over Duggan

Claims 1, 9 and 18

In the paragraph bridging pp. 9 and 10, Applicant states:

Applicants assert that Duggan fails to teach, *inter alia*, "... each of the plurality of instances of the test application run within a single process". In support of the rejection, the Office cites col. 21, lines 53 - 57 in Duggan which provides "...[a] basic module 12 is also responsible for initiating multiple, concurrent sessions". In Duggan, the basic module 12 is but one of a "... plurality of separate Visual Basic code modules ..." of a core module of the test tool program (col. 21, lines 41 - 45) responsible for "... initiating concurrent sessions on different client connections..." However, Duggan fails to disclose that concurrent sessions run in a single process. Instead, the remaining Visual Basic modules come in to play in carrying out the various aspects of the test tool. To this extent, Duggan's basic module's responsibility is not equivalent to the claimed method step of "instantiating a plurality of instances ...using threads, wherein each of the ... instances run within a single process".

(emphasis in original)

Examiner respectfully disagrees. The Authoritative Dictionary of IEEE standards Terms 7th provides the following definition of a process:

process ...

(6) An address space with one or more threads executing within that address space, and the required system resources for those threads. A process is created by another process issuing the *fork()* function. The process that issues *fork()* is known as the parent process, and the new process created by the *fork()* is known as the child process. ...

Duggan's disclosure at col. 21, lines 53-57 ("The basic module 12 is also responsible for initiating multiple, concurrent sessions") appears, on its face, to describe this single process multi-thread situation.

Further Applicant's specification par. [0024] states:

This is in stark contrast to previous methods in which multiple processes were required to instantiate multiple instances of an application. For example, referring to

Fig. 3, the concept of the previous method is shown. As depicted, for each instance of application 52A-B that is instantiated by test client 56, a separate process 58A-B and instance of testing program 60A-B must be provided on test client 56. Since each process 58A-B can consume 10-20 megabytes of memory, the method shown in Fig. 3 is costly and inefficient.

Here it can be seen that the prior art "multi-process" method required multiple instances of the testing program. It should be clear that Duggan's disclosure does not describe a separate instance of basic module 12 for each thread (concurrent session). Accordingly one of ordinary skill in the art would recognize that Duggan's disclosure does not describe the "multi-process" methods to which Applicant refers.

Still further, Applicant's argument that Duggan's 'basic module 12 is but one of a "... plurality of separate Visual Basic code modules ..." does not speak to the number of process used. Those of ordinary skill in the art will recognize that nearly all modern applications implement multiple modules and only some applications create multiple processes.

Claim 2.

In the first full paragraph on pg. 10, Applicant states:

With further respect to dependent claim 2, Applicants respectfully assert, in addition to the above arguments, that Duggan also fails to teach, *inter alia*, "... identifying application protocol interfaces (APIs) ..., prior to the instantiating step...". Claim 2. The Office cites col. 12, lines 21 - 23 in Duggan, which provides "A list box 272 ...all the commands in command module for ...testing a given application program" in support of its rejection. As defined in col. 5, lines 57 - 62, "A 'command' is a series of program instructions which ...cause ...the test tool program to perform a use function of the web application under test via a client connection" but does not disclose or explicitly teach the treatment of application protocol interfaces (APIs). In contrast to Duggan, the claimed invention discloses a method step for "identifying APIs ...prior to instantiating step". To this extent, Duggan teaches a command module with commands for creating a test script but does not teach identifying APIs.

(emphasis in original)

Examiner respectfully disagrees. First it is noted, as discussed below, that Examiner asserts Duggan's 'command modules' and the associated 'commands' (e.g. col. 3, lines 31-46) anticipate Applicant's claimed API's.

Further, Duggan discloses "The core module of the program is independent of the command module; ... and does not have to be rewritten for different application programs to be tested." (col. 3, lines 19-22); "different command modules can be created for different application programs to be tested" (col. 3, lines 31-33) and "A list box 272 contains a list of all of the commands in the command module created for testing a given application program" (col. 12, lines 21-23).

As noted in the rejection, one of ordinary skill in the art would recognize the if the core module has no 'hard-coded' knowledge of (i.e. is independent of) the command module created for particular application program (see col. 3, lines 31-33), populating a list box with the commands of that particular application program (see col. 12, lines 21-23) necessarily requires an identification and loading of the command module associated with that application program (i.e. retrieving the command module to parse out the command names to display in the list box).

In the paragraph bridging pp. 10 and 11, Applicant states:

Additionally, Applicants further assert that Duggan also does not teach, *inter alia*, "providing a test script capable of invoking the APIs ..." In support of the rejection, the Office states that "[the] test operator" in col. 13, lines 59 - 62 in Duggan is equivalent to the claimed "... a test script ...". However, Duggan's test operator teaches the "[creation of] test scripts containing command module commands" but does not teach the invocation of APIs. As such, the claimed test script that is capable of invoking the APIs is not equivalent to Duggan's test operator that is

capable of creating test scripts with command module. Accordingly, Applicants request that the Office withdraw the rejection of claim 2.

Examiner respectfully disagrees. First it is noted that Duggan's 'test operator' is a user of the system who *creates* the claimed "test script" and *not* the test script itself. (see col. 13, lines 59-62 'a test operator [can] create test scripts"). As noted in the rejection it is Duggan's 'test scripts' (col. 13, lines 59-62) which are cited as anticipating Applicant's claimed 'test script'.

Further, it is Duggan's 'command modules' and associated 'commands' (see col. 2, lines 65-67 "The command module contains a number of different commands") which are cited as anticipating Applicant's claimed invoked API's. Specifically those of ordinary skill in the art would recognize that Duggan's test scripts, which are comprised of "command module commands" (13, lines 59-62) are 'capable of invoking the APIs' (col. 13, lines 46-50 "Each of the commands ... cause the computer ... to perform a user function of the application program").

Still further Applicant has not provided any indication how the claimed "invoking the APIs" is distinguished from a script executing Duggan's 'commands' (col. 13, lines 46-50).

For the reasons given above, Applicant's arguments regarding the rejection of claims 1-5, 7-14, 16-23 and 25-26 over Duggan are not persuasive and are maintained.

Rejection of Claims 6, 15 and 24 over Duggan in view of Lindholm

In the paragraph bridging pp. 11 and 12, Applicant states:

Specifically, the Office states that it would have been obvious to a person of ordinary skill in the art to implement Duggan's 'test-tool' and 'basic module' in JAVATM language/programs and execute them on a JVM. However, the Office has not established a rationale or suggested any guidelines that a person of ordinary skill in the art would follow in implementing the teachings of Duggan in Lindholm's JVM. Duggan's 'test-tool' is implemented through multiple Visual Basic code modules collectively comprised within a core module, where the 'basic module' as one of a plurality of Visual Basic code modules. Col. 21, lines 30 - 46. ... Applicants assert that the Office has relied upon hindsight to derive a motivation for a person of ordinary skill in the art to combine the teachings of Duggan and Lindholm. Applicants respectfully submit that the suggestion/motivation to combine the teachings of two references has to originate from the teachings themselves or the knowledge in the art at the time of the invention and not hindsight, which is impermissible.

Accordingly, Applicants respectfully request that the Office withdraw its rejection.

Respectfully, in col. 27, lines 12-15 Duggan discloses "the present invention is not limited to the particular embodiments disclosed, but is intended to cover all modification that are within the spirit and scope of the invention as defined by the appended claims".

Nowhere in his claims does Duggan require the use of Visual Basic (or any other specific language) to implement the multiple modules, which collectively comprise the disclosed core module. Accordingly those of ordinary skill in the art would recognize that Duggan's disclosed system is not tied to implementation in any specific language, but rather is directed to the disclosed interaction of the various objects.

Further, those of ordinary skill in the art would recognize that it would have been a simple matter to implement Duggan's 'plurality of Visual Basic code modules' (col. 21, lines 30-46) as a plurality of JAVA objects by using JAVA code in place of Visual Basic code, and that by doing so, they could achieve many of the benefits of the JAVA language taught by Lindholm (e.g. pg. 1, par. 1).

Accordingly it can be seen that Examiner's asserted motivation to modify the Duggan reference is found in the teachings of both Duggan (e.g. col. 27, lines 12-15) and Lindholm (e.g. pg. 1, par. 1) and thus does not rely upon improper hindsight.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-5, 7-14, 16-23 and 25-26 are rejected under 35 U.S.C. 102(b) as being anticipated by US 6,002,871 to Duggan et al. (Duggan).

Regarding Claims 1, 9 and 18: Duggan discloses providing a test application that satisfies reentrancy requirements (col. 21, lines 57-61 'Each session is ... reentrant') on a client (col. 5, lines 18-21 'the test tool ... runs on a single computer'); and instantiating a plurality of instances of the test application using threads (col. 21, lines 57-61 'Each session is executed as a separate thread'), wherein each of the plurality of instances of the test application run within a single process (col. 21, lines 53-57 'The basic module 12 is also responsible for initiating multiple, concurrent sessions').

Note that Duggan's 'test application' satisfies reentrancy requirements (col. 21, lines 57-61 'the reentrant nature of the test tool ... allows multiple sessions') and thus satisfies the language recited in claims 9 and 18.

Further note that because Duggan's 'list box 272' displays only the commands from 'the command module created for testing a given application' that particular 'command module' must have been identified to the 'list box 272'.

Regarding Claim 2: The rejection of claim 1 is incorporated; further Duggan discloses identifying application protocol interfaces (APIs) associated with the test application prior to the instantiating step (col. 12, lines 21-23 'A list box 272 contains a list of all of the commands in the command module created for testing a given application program'); and providing a test script capable of invoking the APIs (col. 13, lines 59-62 'a test operator [can] create test scripts containing ... command module commands'), wherein upon execution, the test script instantiates the plurality of instances of the test application (col. 5, line 67-col. 6, line 3 'the test tool program executes multiple concurrent sessions') using threads (col. 21, lines 57-61 'Each session is executed as a separate thread') within the single process (col. 21, lines 53-57 'The basic module 12 is also responsible for initiating multiple, concurrent sessions').

Regarding Claims 3, 14 and 23: The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses the server application is a network application (col. 5, lines 9-12 'a test tool for testing application programs ... over a network').

Regarding Claims 4, 12 and 21: The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses the reentrancy requirements dictates that the plurality of instances of the test application be run within the single process without interfering with each other (col. 21, lines 57-61 'reentrant nature of the test tool').

Regarding Claims 5, 13 and 22: The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses each of the plurality of instances of the test application corresponds to a separate thread (col. 21, lines 57-61 'Each session is executed as a separate thread'), and wherein each of the separate threads is associated with a different connection to the server (col. 5, line 66-col. 6, line 3 'A "session" refers to the execution of one test script, on one client connection').

Regarding Claims 7, 16 and 25: The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses the plurality of instances of the test application simulate use of the server application by a plurality of users (col. 6, lines 47-51 'the test tool program ... is capable of executing test scripts ... based on a user list').

Regarding Claims 8, 17 and 26: The method of claim 1, 9 and 18 further comprising collecting data corresponding to the test (col. 8, lines 4-6 'The test tool program ... provides four options for logging information').

Regarding Claims 10 and 19: The rejection of claims 10, and 19 are incorporated respectively, further; Duggan discloses an interface identification system for identifying application protocol interfaces (APIs) associated with the test application (col. 12, lines 21-23 'A list box 272 contains a list of all of the commands in the command module created for testing a given application program').

Regarding Claims 11 and 20: The rejection of claims 10, and 19 are incorporated respectively, further; Duggan discloses the test application instantiation system comprises a driver that executes a test script capable of invoking the identified APIs (col. 13, lines 59-62 'a test operator [can] create test scripts containing ... command

module commands), and wherein upon execution, the test script instantiates the plurality of instances of the test application (col. 5, line 67-col. 6, line 3 'the test tool program executes multiple concurrent sessions') using threads (col. 21, lines 57-61 'Each session is executed as a separate thread') within the single process (col. 21, lines 53-57 'The basic module 12 is also responsible for initiating multiple, concurrent sessions').

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 6, 15 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,002,871 to Duggan et al. (Duggan) in view of “The Java™ Virtual Machine Specification” by Lindholm et al (Lindholm).

Regarding Claims 6, 15 and 24: The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan does not disclose the process comprises a JAVA virtual machine.

Lindholm teaches that JAVA programs, which run on a JAVA virtual machine were well known at the time of the invention, and that JAVA programs and the JVM provided benefits known to those of ordinary skill in the art. Accordingly it would have been obvious to a person of ordinary skill in the art at the time of the invention to implement

Duggan's 'test tool' and 'basic module' in the JAVA programming language and execute them on a JVM.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Mitchell whose telephone number is (571) 272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



Jason Mitchell
10/31/06



MENG T. AN
SUPERVISORY PATENT EXAMINER
COMPUTER CENTER 2100